Copilot Arena: A Platform for Code LLM Evaluation in the Wild

Wayne Chi* , Valerie Chen*
Anastasios Nikolas Angelopoulos, Wei-Lin Chiang, Aditya Mittal, Naman Jain, Tianjun Zhang, Ion Stoica, Chris Donahue, Ameet Talwalkar

Motivation

Static benchmarks do not have users in the loop	**User studies** operate on a limited, prescribed set of tasks	**Preference evaluations** do not occur in realistic coding envs

Limitations of existing evaluations

System Design

Copilot Arena is a VSCode extension that provides users with pairs of inline code completions from various LLMs. Users provide their votes on which completion is better suited for their task.

Model Sampling

- We optimize the trade-off between a latency-optimized distribution and a uniform distribution (i.e., to improve coverage).
- This approach decreased median experienced latency by 33%.

Model prompting

- On HumanEval-Infilling, many chat LLMs struggle to "fill in the middle" (FiM).
- Allowing LLMs to generate code snippets and post-processing them into a FiM completion improves performance 93% of the time.

Leaderboard

	Copilot Arena	LiveBench	BigCodeBench	LiveCodeBench	Chatbot Arena (general)	Chatbot Arena (coding)
deepseek-coder	1	-4	-5	-	-1	0
claude-3.5-sonnet	1	0	-2	-1	-4	0
codestral	2	-	-	-6	-	-
llama-3.1-405b	3	-4	-3	-1	-2	-1
gemini-flash-002	3	-5	-	-4	-1	-6
gemini-pro-002	3	-1	-2	-3	+2	0
gpt-4o-2024-08-06	4	+1	+3	+3	-3	-3
llama-3.1-70b	4	-5	0	-5	-4	-2
qwen-2.5-coder-32b	9	+7	+8	+6	0	+1
gpt-4o-mini	9	+3	+1	+4	+6	+5
		r=0.10	r=-0.15	r=-0.10	r=0.48	r=0.62

Copilot Arena

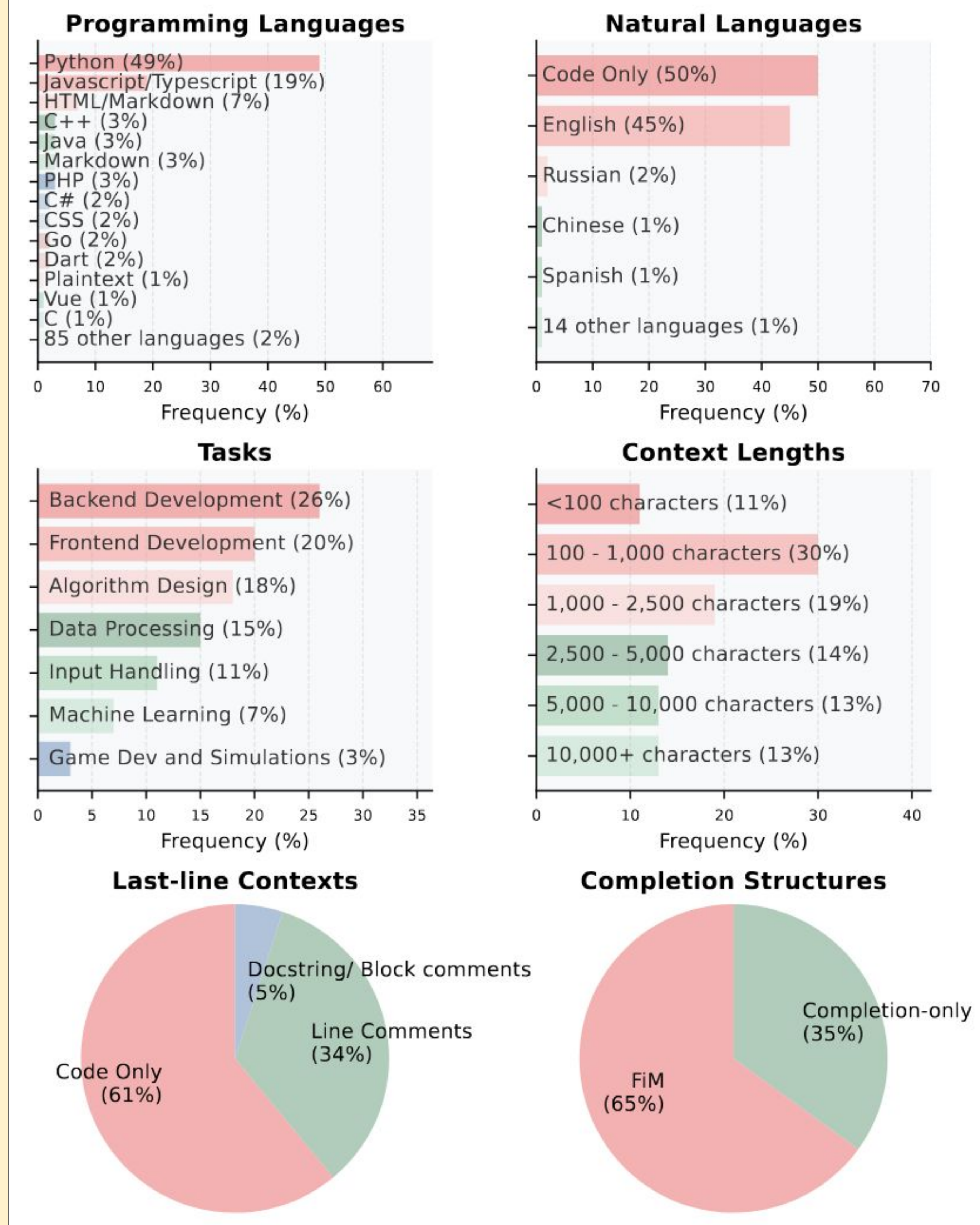
has served over 4.5 million suggestions from 10 LLMs and collected over 11k votes.

We find the following key insights:

1. Existing evaluations do not necessarily correlate well with in-the-wild preferences.
2. Model performance is affected by task, context, and code structure. No model that is "one-size-fits-all."
3. Diverse and realistic human preference data is essential for effective code generation models. Smaller models tend to perform better on data similar to static benchmarks.

Full Paper **Repository**

Data Analysis



Key differentiators in our data:

- 103 programming languages; significantly more than other benchmarks
- Less interview style problems (i.e., algorithm design) and more real-world problems (i.e., backend and frontend development).
- Structurally diverse problems, comprising a mixture of infilling versus code completion and forms of docstring tasks.

Insights into user preferences

We partition each feature into contrasting subsets (e.g. FiM vs non-FiM). We compute a win-rate difference matrix, i.e., the number of substantial differences in the win-rate between each subset:

	Front/Backend	Long Context	FiM	Non-Python
deepseek-coder	0, -3	+2, 0	+1, 0	0, 0
claude-3.5-sonnet	+4, 0	0, -1	+2, 0	+1, 0
codestral	+1, 0	+1, -1	0, 0	0, 0
llama-3.1-405b	+1, -4	+1, -1	0, 0	0, 0
gemini-flash-002	+1, -2	0, 0	+1, -2	0, 0
gemini-pro-002	+1, 0	+3, 0	+2, 0	0, -1
ɟpt-4o-2024-08-06	+1, 0	0, -2	0, -2	+1, 0
llama-3.1-70b	+4, 0	+1, 0	+1, -2	0, 0
ven-2.5-coder-32b	0, -2	0, -3	0, 0	0, -2
gpt-4o-mini	+1, -3	0, 0	0, -1	+1, 0
% Total Changes:	31.1	17.8	15.6	6.7